Surfel Based Geometry Resonstruction

Andersen, Vedrana; Aanæs, Henrik; Bærentzen, Jakob Andreas

Published in:
Theory and Practice of Computer Graphics 2010

Publication date:
2010

Document Version
Early version, also known as pre-print

EG UK Theory and Practice of Computer Graphics (2010)
John Collomosse, Ian Grimstead (Editors)

Surfel Based Geometry Reconstruction

Vedrana Andersen[1], Henrik Aanæs[1] and Andreas Bærentzen[1]

[1]Technical University of Denmark

Abstract

We propose a method for retrieving a piecewise smooth surface from noisy data. In data acquired by a scanning process sampled points are almost never on the discontinuities making reconstruction of surfaces with sharp features difficult. Our method is based on a Markov Random Field (MRF) formulation of a surface prior, with the surface represented as a collection of small planar patches, the surfels, associated with each data point. The main advantage of using surfels is that we avoid treating data points as vertices. MRF formulation of the surface prior allows us to separately model the likelihood (related to the mesh formation process) and the local surface properties. We chose to model the smoothness by considering two terms: the parallelism between neighboring surfels, and their overlap. We have demonstrated the feasibility of this approach on both synthetical and scanned data. In both cases sharp features were precisely located and planar regions smoothed.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

1. Introduction

In this paper, we propose a novel anisotropic method for smoothing noisy data. We represent the surface using small planar patches, the *surfels*, associated with each data point. This allows us to easily define a surface prior based on Markov Random Fields (MRFs).

Markov Random Fields have been used extensively for solving Image Analysis problems at all levels. While some examples are mentioned below, MRFs have rarely been used for mesh processing. The central element of the MRF formulation is that we use Bayes' rule to express the probability of a given field (in this case a surface) as the product of a *likelihood* and a *prior*. Likelihood relates to our knowledge of the noise (e.g. how much noise a scanner introduces), while the prior relates to our knowledge of the properties of the surface (e.g. how smooth a surface should be).

Representing a surface using surfels has some clear advantages. To begin with, surfel representation corresponds well to the data creation process, as each sampled point corresponds to the scanner detecting the objects surface, and is rarely, if never, the point on the sharp feature. By using surfels we also avoid the problems of dealing with sometimes arbitrary triangulation, where the mesh edges correspond poorly with the sharp features on the surface.

2. Related Work

A very important task in geometric processing, and a main way of generating 3D content, is the estimation of 3D shape or geometry from observations. In many of these cases mesh smoothing are needed to reduce noise in the data, resulting in a variety of proposed algorithms, from the early isotropic [DMSB99, Tau95], over anisotropic [DMSB00, TWBO02], and efficient feature preserving [DTB06, FDCO03, JDD03] methods.

The most popular approach is curvature minimization based smoothing. However, this is not suitable in all cases. An example is man made environments where the geometry often is piecewise planar. This nature does not correspond well with curvature minimization, since the curvature is high (in principle infinite) at the corners. A way of addressing this problem is to find the most dominant planar directions in the data and constraining the data to this [FCSS09a, FCSS09b].

To allow for piecewise smooth surfaces using a local approach, a number of smoothing schemes employ smoothing of the normal and then reconstructing the point location [SB04, SRML07]. The main drawback of most of the proposed methods is that they depend on having samples on discontinuities, or try to migrate vertices to the sharp features.

Our work solves the issue of expecting vertices to be on the feature edges by using surfel surface representation. Surfels (surface elements with no connectivity information) has been introduced in [PZvBG00] as primitives for rendering, in an extension of point rendering [GD98]. We propose using surfels for 3D surface estimation using a MRF [Li01] formulation of piecewise planar prior. A similar approach has been tried [AABN10], but by using surfel surface representation our work solves the issue of expecting vertices to be on the feature edges.

Dual mesh approach, which we apply, is used in [OB02] in the context of optimizing isosurface polygonization, noticing the advantage of using dual when recovering sharp features.

Comprehensive study on the use of MRF theory for solving image analysis problems can be found in books by [Li01] and [Win03]. MRF theory is convenient for addressing the problem of piecewise smooth structures. In [GG84] a foundation for the use of MRF in image analysis problems is presented in an algorithm for restoration of piecewise smooth images, where gray-level process and line processes are used. Some of the other applications of MRFs for problems involving reconstruction of piecewise smooth structures include [DT05], where high-resolution range-sensing images are reconstructed using weights obtained from a regular image. In [HC03] a coupled MRF is used for locating grids with possible cracks in the structure. In [SSZ03] a stereo matching problem is addressed by three coupled MRFs modeling piecewise smoothness and occlusion.

There are some previous examples of using MRF theory to 3D meshes. In [WSC04] MRF are used in the context of surface sculpting with the deformation of the surface controlled by MRF potentials modelling elasticity and plasticity. MRF was also used for mesh analysis and segmentation in [LW08], point cloud reconstruction in [JWB*06], and surface reconstruction [PBL09].

3. Markov Random Fields Theory

MRFs is a powerful framework for expressing statistical models originating in computational physics, and it has proven highly successful in image analysis. A MRF is, essentially, a set of sites with associated labels and edges connecting every site to its neighbors. The labels are the values which we wish to assign (e.g. pixel color, vertex position, surfel parameters), and it is a central idea in MRF theory that the label at a given site must only depend on the labels of its neighbors.

Apart from a well developed mathematical framework one of the main advantages of MRF is that its Markovianity (local property) makes it quite clear what the objective function is and what a MRF based algorithm aims at achieving. Exponential distributions are often used, and the joint probability distribution function of given configuration f (e.g. combined vertex location) is given by

$$P(f) \propto e^{-\sum U(f)} \ ,$$

where the $U(f)$ can be seen as energy terms or potentials defined on neighborhoods. In order to find the most likely configuration f, we need to obtain

$$\min_f \sum U(f) \ . \tag{1}$$

In our proposed framework, we wish to smooth a set of surfels. Some of the $U(f)$ in (1) are therefore data (likelihood) terms penalizing the displacement from the input mesh. Other terms would be prior terms which express how likely a surface is *a priori*.

4. The Method

The input to our method is a triangle mesh. Since we use the mesh connectivity only to efficiently find neighbors of each data point, the method can easily be modified to take a point cloud as input.

Starting from the noisy input mesh, we associate a planar patch to each vertex. We formulate a surface prior by defining two terms, the parallelism term and the overlap term. Parallelism depends on the orientations of the surfels, while the overlap depends on the distance between the surfels. Both terms are weighted to account for sharp edges – the discontinuities between the smooth parts of the surface.

In the iterative scheme we optimize the parallelism and the overlap between neighboring surfels. The weighting is gradually increasing, to precisely detect the sharp edges and impose smoothness. Finally we retrieve the piecewise smooth surface by robustly estimating the intersections between neighboring surfels.

4.1. Surfel Representation

Our surfel representation associates each data point with a small planar patch, a *surfel*. A surfel $(\mathbf{n}, a)$ is a piece of a plane and is represented by a plane normal $\mathbf{n}$ and a distance a from the origin.

Unlike planes, surfels are localized. A center $\mathbf{c}$ of the surfel is the projection of the data point $\mathbf{p}$ onto the surfel plane

$$\mathbf{c} = \mathbf{p} - (\mathbf{n} \cdot \mathbf{p} - a) \cdot \mathbf{n} \ ,$$

where $(\ \cdot\ )$ denotes a scalar product.

In our implementation the input was a noisy triangular mesh, so we assigned a surfel to each of the mesh vertices v.

4.2. Surfel Initialization

Surfels are initially positioned in the tangential plane of each data point. When associating a surfel to a mesh vertex v we have

$$v \mapsto (\mathbf{n}, a) = (\mathbf{n}, \mathbf{n} \cdot \mathbf{p}) \ ,$$

where $\mathbf{n}$ is some normal estimate at v, and $\mathbf{p}$ is the position of v.

We utilized the mesh connectivity and used area-weighted normal as the estimation of the surface normal, but any other normal estimation would perform just as well. The final optimization result proved rather robust to initialization when we experimented with adding a small amount of noise to the initial normal estimate.

4.3. Surfel Optimization

Smoothing of the surfels is done by minimizing the objective function consisting of three parts: likelihood, parallelism and overlap. Each part can be formulated as the energy contribution of each surfel, or as a joint energy for the whole mesh, which is simply a sum of the individual contributions. For simplicity, and because it corresponds well to our optimization scheme, we formulate the objective function of a single surfel.

In our optimization, we visit every surfel and locally adjust its parameters according to the objective function. We repeat until the energy converges, which generally happens after only a few iterations.

Likelihood. It is possible to utilize the knowledge about the data acquisition process (e.g. scanner accuracy and geometry) by formulating a suitable likelihood energy. Likelihood expresses the probability that a given surface is a corrupted version of some other surface.

Since the likelihood was not the object of our experiments, we assumed the isotropic Gaussian noise. The contribution of the single surfel to the likelihood is a squared distance between the surfel $(\mathbf{n}, a)$ and the data point $\mathbf{p}$

$$(\mathbf{n}\cdot\mathbf{p}-a)^2 \ .$$

The other two contributions to the objective function, the parallelism and the overlap, form a smoothness prior, which expresses how probable (a priori) a given surface is. The prior encodes the belief that a smooth surface is more probable than a noisy surface, and in MRF framework prior is defined locally by penalizing the undesired behavior of the surface. In the case of our smoothness prior, both the parallelism and the overlap are formulated for the pairs of neighboring surfels.

In our implementation we have used the mesh connectivity to determine the set of neighboring vertices. This can however easily be replaced by the neighboring relation defined via spatial proximity, meaning that the basic algorithm can be adopted to take a point cloud as input.

Parallelism. Difference in the orientation of two neighboring surfels is penalized by the parallelism term. For every pair of neighboring surfels $(\mathbf{n}_i, a_i)$ and $(\mathbf{n}_j, a_j)$ the contribution to the parallelism term is the squared distance between the normals

$$\|\mathbf{n}_i-\mathbf{n}_j\|^2 \ .$$

The parallelism energy corresponding to a each surfel is therefore

$$\sum_{j\in N(v)} \|\mathbf{n}-\mathbf{n}_j\|^2 \ ,$$

where $N(v)$ denotes indices of the vertices neighboring to v.

Overlap. Parallelism term will ensure the smoothness of the normal field but we also want to adjust the drift of each surfel in the normal direction so that the resulting surface is continues. To assure an appropriate overlap between the neighboring surfels we penalize the local distance between surfels.

For two sufels $(\mathbf{n}_i, a_i)$ and $(\mathbf{n}_j, a_j)$ we consider the length of the projection of surfel center $\mathbf{c}_j$ to a plane of the surfel $(\mathbf{n}_i, a_i)$

$$(\mathbf{n}_i\cdot\mathbf{c}_j-a_i)$$

and vice versa. The sum of the squared distances is the term we want to minimize to obtain the local overlap between the surfels.

As a result, each surfel contributes with the overlap energy

$$\sum_{j\in N(v)} \left((\mathbf{n}\cdot\mathbf{c}_j-a)^2+(\mathbf{n}_j\cdot\mathbf{c}-a_j)^2\right) \ .$$

Weights. Minimizing the prior term as defined above will smooth the surfels. However, we are interested in retrieving a piecewise smooth surface, so we do not want to penalize all sharp edges and corners. To allow for the surface to break across sharp ridges we are using adaptive thresholding weights. When considering the optimal function for each surfel we start by looking at the parallelism energy for all of the neighbors.

We want to smooth the normals of the surfels, but not if the angle between the normals is larger than the certain threshold. Therefore we obtain weights from the parallelism energy using a thresholding function

$$f(x)=\frac{1}{1+(\frac{x}{t})^2} \ ,$$

where t is a threshold parameter. The behavior of the function is illustrated in Fig. 1, *left*. Those weights are then applied both to parallelism term and the overlap term. The effect of thresholding the parallelism term is illustrated in Fig. 1, *right*.

Iterations. Our optimization scheme is iterative. In each iteration we visit all the surfels and adjust its parameters to minimize the objective function. In our experience, the convergence is achieved after only a few iterations. In order

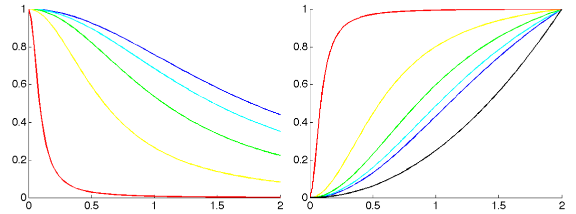

Figure 1: *Thresholding function used to achieve feature preserving behavior of the smoothness prior.* Left: *Thresholding function for the different threshold values, blue corresponds to 125 degrees angle, while the red corresponds to an angle of 5 degrees.* Right: *The effect of the weighting parallelism energy with the thresholding function for the same threshold values. The black line shows parallelism energy without weighting*

to efficiently smooth the planar parts of the mesh, and still preserve the sharp features, we are moving the threshold of the weighting function with each iteration. As a result two neighboring surfels are either almost parallel, or at an angle, which is so sharp that the thresholding function effectively breaks all smoothing.

Our current implementation is based on the Matlab's numerical optimization, which allowed us to experiment with a number of different priors without adjusting the optimization scheme.

4.4. Surface Retrieval

The result of the surfel optimization is a collection of small surface patches. Finding a suitable and robust method for final retrieval of the surface is still a part of our ongoing work. To visualize and verify the results we have utilized the initial mesh connectivity.

As we want the final result to reflect the fact that we associate the surface patch to a data point, directly returning to initial mesh connectivity would be inappropriate. Instead, we have calculated a dual mesh, associating the dual face with each vertex of initial mesh, and a dual vertex to each face of initial mesh. The key element of our approach is calculating the positions of dual vertices.

Each of the dual vertices, corresponding to the face of the initial mesh, has three neighboring faces and the associated surfels. The position of the dual vertex was calculated as a robust intersections of the three neighboring surfel planes. We used use quadric error metrics [GH97, GH98] to find the plane intersection. Error quadrics can be used to minimizes the squared distance of the point to the set of planes, keeping track whether a single plane is counted multiple times.

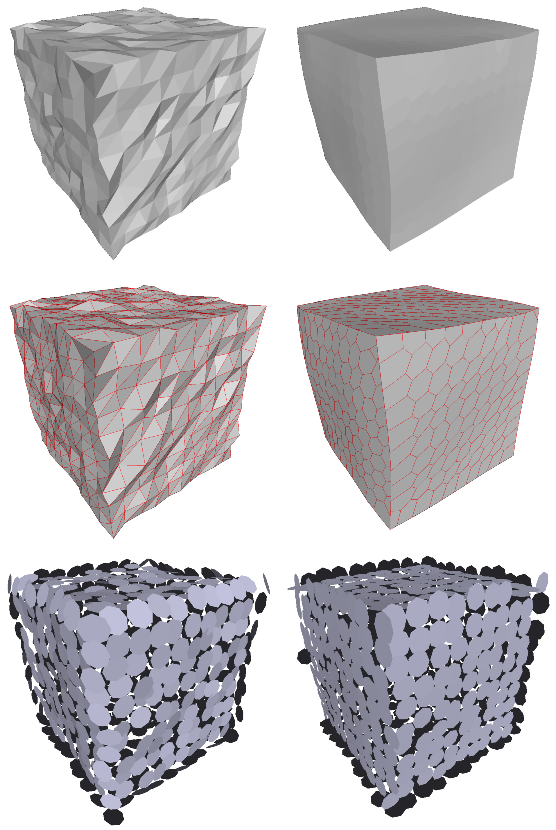

Figure 2: *The reconstruction of the synthetic cube corrupted by Gaussian noise.* Left: *Two renderings of the input mesh and the initial surfel configuration.* Right: *Corresponding renderings of the reconstructed surface and optimized surfel configuration*

5. Results

Our method can retrieve piecewise smooth surfaces from rather noisy data. It is also evident that the sharp features are recovered with great accuracy.

In Fig 2 we show the result of our initial experiments, where we smooth the synthetic cube corrupted with a Gaussian noise. We have also tested our method on a broadly used fandisk object, see Fig 3.

The main application of our method is to reconstruct the piecewise smooth surface from the noisy scans. To perform test on a real life example, we scanned a squared block in a structured light scanner. The results are shown in Fig 4 (note that the ragged ends are the boundaries of the scan).

6. Conclusion

We have developed a method for reconstructing piecewise smooth surfaces from scanned data. Our method uses a sur-

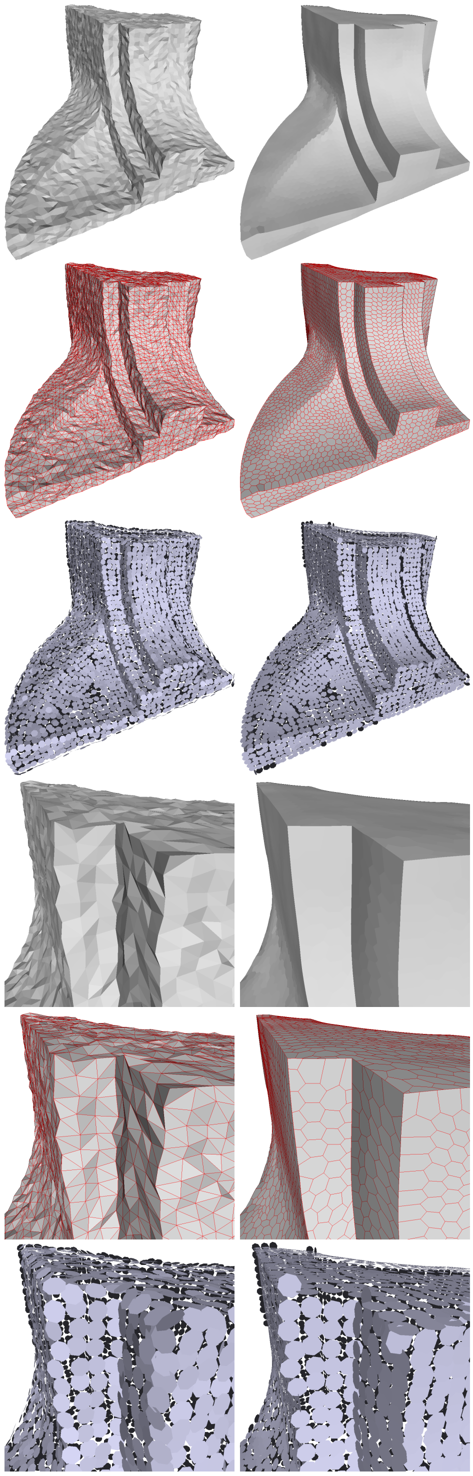

Figure 3: *The reconstruction of the fandisk model corrupted by the Gaussian noise.* Left: *Input mesh and the initial surfel configuration. On the bottom a close up of the sharp detail.* Right: *The corresponding views on the reconstructed surface and the optimized surfel configuration*

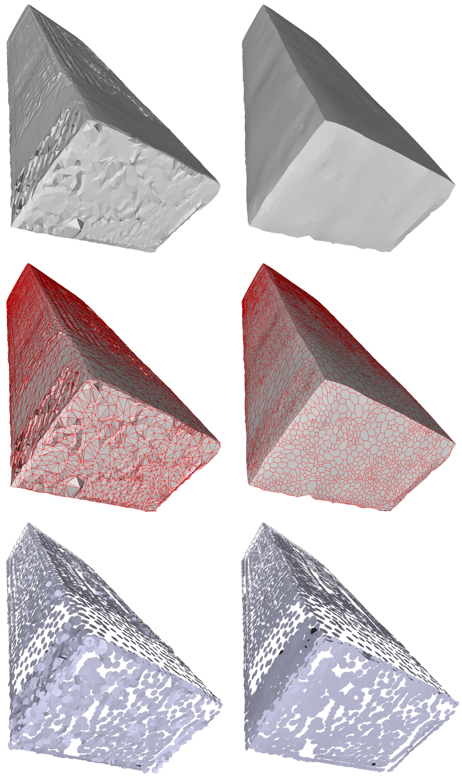

Figure 4: *A structured light scan of a cubical model and its reconstruction.* Left: *Two renderings of the input mesh (displaying both the irregularly sampled points, scanning noise and the triangulation artifacts) and the initial surfel configuration.* Right: *The corresponding renderings of the reconstructed surface and the optimized surfel configuration. Note that the ragged edges scan boundaries.*

fel representation of surfaces. We assign plane parameters to each of the scanned points, and adjust those parameters according to the smoothness prior. The main contribution of our method is that we show the advantage of not assuming samples on discontinuities. Instead we are optimizing pieces of smooth surfaces resulting in well reconstructed piecewise smooth geometry. Using a dual mesh while reconstructing the surface we again avoid the issue of missing sample points on sharp features. Our experiments demonstrate the capability of our method.

References

[AABN10] ANDERSEN V., AANÆS H., BÆRENTZEN J. A., NIELSEN M.: Markov random fields on triangle meshes. In *WSCG'10: International Conference on Computer Graphics, Visualization and Computer Vision* (2010), no. 18, pp. 265–270. 2

[DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press, pp. 317–324. 1

[DMSB00] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Anisotropic feature-preserving denoising of height fields and images. In *Proceedings of Graphics Interface* (2000), pp. 145–152. 1

[DT05] DIEBEL J. R., THRUN S.: An application of Markov random fields to range sensing. In *Proceedings of Conference on Neural Information Processing Systems* (2005). 2

[DTB06] DIEBEL J. R., THRUN S., BRÜNIG M.: A Bayesian method for probable surface reconstruction and decimation. *ACM Transactions on Graphics 25*, 1 (2006), 39–59. 1

[FCSS09a] FURUKAWA Y., CURLESS B., SEITZ S. M., SZELISKI R.: Manhattan-world stereo. *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (2009), 1422–1429. 1

[FCSS09b] FURUKAWA Y., CURLESS B., SEITZ S. M., SZELISKI R.: Reconstructing building interiors from images. *International Conference on Computer Vision* (2009). 1

[FDCO03] FLEISHMAN S., DRORI I., COHEN-OR D.: Bilateral mesh denoising. In *SIGGRAPH '03: ACM SIGGRAPH Papers* (2003), ACM Press, pp. 950–953. 1

[GD98] GROSSMAN J., DALLY W. J.: Point sample rendering. In *Rendering techniques' 98: proceedings of the Eurographics Workshop in Vienna, Austria* (1998), Springer Verlag, p. 181. 2

[GG84] GEMAN S., GEMAN D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 6*, 332 (1984), 721–741. 2

[GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), ACM Press, p. 216. 4

[GH98] GARLAND M., HECKBERT P. S.: Simplifying surfaces with color and texture using quadric error metrics. In *VIS '98: Proceedings of the conference on Visualization '98* (1998), IEEE Computer Society Press, pp. 263–269. 4

[HC03] HARTELIUS K., CARSTENSEN J. M.: Bayesian grid matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 2 (2003). 2

[JDD03] JONES T. R., DURAND F., DESBRUN M.: Non-iterative, feature-preserving mesh smoothing. In *SIGGRAPH '03: ACM SIGGRAPH Papers* (2003), ACM Press, pp. 943–949. 1

[JWB*06] JENKE P., WAND M., BOKELOH M. S., SCHILLING A., STRASSER W.: Bayesian point cloud reconstruction. In *Computer Graphics Forum* (2006), vol. 25, pp. 379–388. 2

[Li01] LI S. Z.: *Markov Random Field Modeling in Image Analysis*, second ed. Springer Verlag, Tokyo, 2001. 2

[LW08] LAVOUÉ G., WOLF C.: Markov Random Fields for Improving 3D Mesh Analysis and Segmentation. In *Eurographics 2008 Workshop on 3D Object Retrieval* (2008). 2

[OB02] OHTAKE Y., BELYAEV A. G.: Dual/primal mesh optimization for polygonized implicit surfaces. In *SMA '02: Proceedings of the seventh ACM symposium on Solid modeling and applications* (2002), ACM, pp. 171–178. 2

[PBL09] PAULSEN R. R., BÆRENTZEN J. A., LARSEN R.: Markov Random Field Surface Reconstruction. *IEEE Transactions on Visualization and Computer Graphics* (2009). 2

[PZvBG00] PFISTER H., ZWICKER M., VAN BAAR J., GROSS M.: Surfels: surface elements as rendering primitives. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), ACM Press, pp. 335–342. 2

[SB04] SHEN Y., BARNER K. E.: Fuzzy vector median-based surface smoothing. *IEEE Transactions on Visualization and Computer Graphics 10*, 3 (2004), 252–265. 1

[SRML07] SUN X., ROSIN P. L., MARTIN R. R., LANGBEIN F. C.: Fast and effective feature-preserving mesh denoising. *IEEE Transactions on Visualization and Computer Graphics 13*, 5 (2007), 925–938. 1

[SSZ03] SUN J., SHUM H.-Y., ZHENG N.-N.: Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 7 (2003), 787–800. 2

[Tau95] TAUBIN G.: A signal processing approach to fair surface design. In *SIGGRAPH '95: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (1995), ACM Press, pp. 351–358. 1

[TWBO02] TASDIZEN T., WHITAKER R., BURCHARD P., OSHER S.: Geometric surface smoothing via anisotropic diffusion of normals. In *VIS '02: Proceedings of the Conference on Visualization* (2002), IEEE Computer Society, pp. 125–132. 1

[Win03] WINKLER G.: *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods*. Springer, 2003. 2

[WSC04] WILLIS A., SPEICHER J., COOPER D. B.: Surface sculpting with stochastic deformable 3d surfaces. In *ICPR '04: Proceedings of the 17th International Conference on Pattern Recognition* (2004), vol. 2, IEEE Computer Society, pp. 249–252. 2